Distilling Prompts at Test-Time for Multimodal Few-Shot Learning

Akash Gupta [1] **Amos Storkey** [1] **Mirella Lapata** [1]

Abstract

In-Context Learning (ICL) has been a well-established paradigm to adapt Large Multimodal Models (LMMs) to novel tasks with minimal supervision. However, the ICL performance of LMMs improves inconsistently with increasing examples due to additional information present in image embeddings, which is irrelevant to the downstream task. To address this, we introduce a meta-learning strategy that distills task-relevant image features into a fixed set of soft prompts, which can be fine-tuned with just a few examples at test time. Further, to facilitate this distillation, we propose an attention-mapper module, integrated in the LLaVA v1.5 architecture, and trained alongside the soft prompts to enable rapid adaptation under low-data conditions. We show that on the VL-ICL Benchmark, our method outperforms ICL and other prompt distillation approaches and boosts the few-shot visual question-answering performance of LMMs.

1. Introduction

Learning to adapt quickly from a few examples is one of the amazing capabilties of human intelligence (Griffiths et al., 2019; Kirsch & Schmidhuber, 2022). Artificial agents like Large Multimodal Models (LMMs), also exhibit this few-shot learning ability by relying on In-Context Learning (ICL), which involves prompting these models with a few input-output examples, without requiring any further training. Although this training-free nature of ICL has led to its rapid adoption across tasks (Huang et al., 2024; Hendel et al., 2023), its underlying mechanism remains ill-understood and its empirical behaviour can be inconsistent. Recent work (Zong et al., 2025) demonstrates that ICL is most effective for large-scale LMMs (~72B parameters), while smaller models ($\leq$7B parameters) often struggle with increasing in-context examples and their performance either plateaus or deteriorates even when extending the context length or giving detailed instructions. (Zong et al., 2025) attributes this limitation to the fact that smaller models struggle with the large number of image tokens in long sequences. As a result, they become confused and perform the task haphazardly or revert to default behaviors, such as drawing from their parametric knowledge, while effectively ignoring the in-context examples.

Building upon this, we hypothesize that effective few-shot adaptation at test time for a task may be compromised by the added information introduced by the image embeddings. As an alternative, we propose to learn a fixed set of *new* embeddings that can be easily finetuned at test time. This idea of task adaptation has gained significant traction in the literature through *prompt tuning* (Lester et al., 2021) which finetunes a set of continuous *soft* prompts while keeping the underlying language model frozen. We introduce an approach for learning new tasks using learnable soft prompts that receive task information from the LLM in the form of loss gradients during finetuning. These gradients update the soft prompts which when fused with the image embeddings are able to distill task-relevant features from them. To facilitate this fusion, we utilize the LLaVA v1.5 architecture (Liu et al., 2024) and propose to replace its MLP projection layer with an attention-mapper that uses a multi-head attention (Vaswani et al., 2017) architecture responsible for extracting relevant task-specific image information.

As the above prompt distillation approach relies on being able to adapt quickly to new tasks at *test time* after seeing only a few examples, we take advantage of previous works (Finn et al., 2017; Ravi & Larochelle, 2017) and formulate this as a *meta-learning* problem. Specifically, we employ the widely known MAML algorithm (Finn et al., 2017) and use its lightweight first-order approximation for training the attention-mapper and soft prompts to distill image features. Our contributions can be summarized as follows:

- We propose an alternative to ICL by meta-learning a fixed set of soft prompts within LMMs through distillation. This can quickly adapt to new tasks at test time on a small number of examples and shows monotonic improvement as examples are increased across varying number of soft prompts.

[1] School of Informatics, University of Edinburgh, Edinburgh, UK. Correspondence to: Akash Gupta <akash.gupta@ed.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

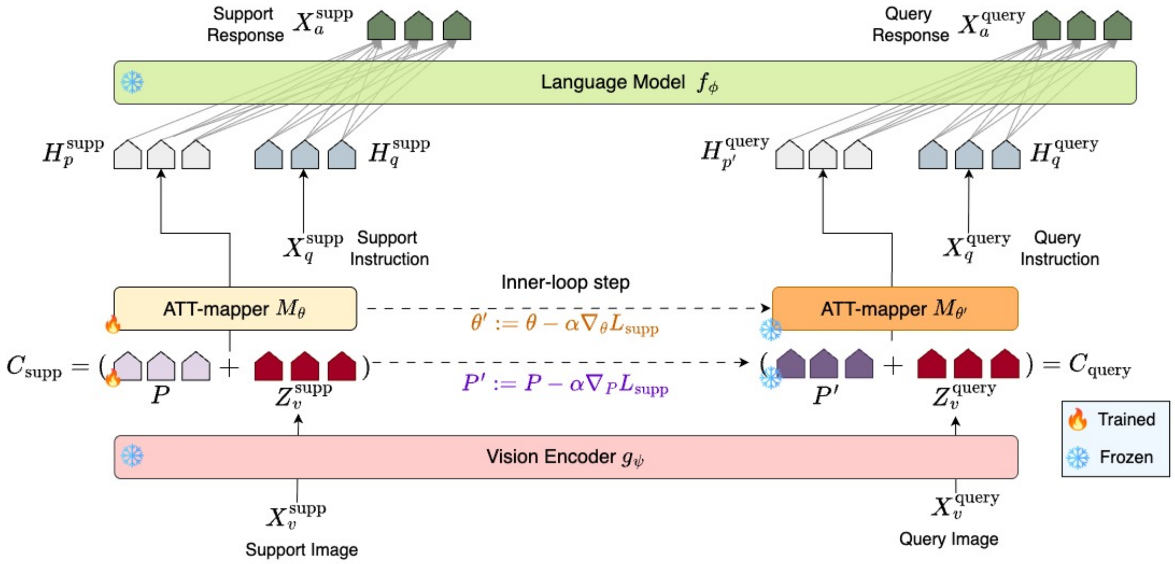


Figure 1. Our proposed meta-learning framework based on LLaVA v1.5-7B (Liu et al., 2024) for distilling image embeddings into soft prompts P during instruction finetuning. First, in the *inner loop*, the support set is processed to obtain loss value L_{supp}, which is then used to obtain task-specific parameters $\{\theta', P'\}$ with a few gradient descent steps. Then, in the *outer loop*, the task-specific parameters are used to infer on the query set to calculate the query loss for meta-parameter optimization $\{\theta, P\}$.

- We incorporate an attention-mapper, inspired from (Najdenkoska et al., 2023), into the LLaVA-v1.5 7B architecture that is trained jointly with *soft* prompts and facilitates the distillation of task-specific image information. We further replace LLaVA's original LLM with a more powerful model, namely Qwen2.5-7B-Instruct (Qwen et al., 2025) to learn better prompts.
- Extensive evaluation on VL-ICL Bench[1] (Zong et al., 2025), a diverse benchmark for image perception and mathematical reasoning, demonstrates that our approach outperforms ICL and other prompt distillation methods.

2. Methods

2.1. Meta-task Creation

We focus on the problem of few-shot visual question answering (VQA), which is derived from the traditional VQA (Antol et al., 2015) setup that contains a dataset $\mathcal{D}$ with corresponding train and test splits $(D^{\text{train}}, D^{\text{test}})$. We maximize the following joint likelihood, $\prod_{i=1}^{|D^{\text{train}}|} p_\theta(X_a^i|X_v^i, X_q^i)$ of answer X_a, given an image X_v and a question X_q over D^{train} during training such that it also maximizes this likelihood on D^{test}. For few-shot VQA, we treat the in-context examples (or shots) given to an LMM during ICL as D^{train}. Since the examples in D^{train} are few (as low as 1-shot), it becomes harder to avoid overfitting while training and still perform well on D^{test}. We conceptualize this problem as one of learning about an underlying task represented by D^{train} and adopt meta-learning (Finn et al., 2017) which exploits the shared structure across a distribution of tasks to learn a prior over model parameters, thereby enabling stable transfer to new tasks with low data.

Optimization-based meta-learning (Finn et al., 2017) involves processing batches of few-shot datasets that represent an underlying task. We start by utilizing the finetuning data mixture of LLaVA datasets (More details in Appendix A.2) to build $\mathcal{D}$ and curate few-shot datasets, which we refer to as *meta-tasks*. Formally, we create a meta-task T^j by randomly sampling a fixed subset of examples from dataset $D^i \sim p(\mathcal{D})$ and partitioning the examples further into support and query sets $T^j = \{D^{\text{supp}}, D^{\text{query}}\}$, where $D^{\text{supp}} \equiv D^{\text{train}}$ and $D^{\text{query}} \equiv D^{\text{test}}$. We continue this process until all samples from D^i have been assigned to at least one meta-task. This meta-task construction is performed for each dataset in $p(\mathcal{D})$, resulting in a meta-task distribution.

2.2. Model Architecture

Figure 1 shows our model architecture which builds on the visual instruction tuning framework of LLaVA v1.5 (Liu et al., 2024). The model consists of a pretrained CLIP ViT-L/14 visual encoder (g_ψ) that processes the input image X_v[2]

[1] We only focus on single-image few-shot visual question answering (VQA) tasks and leave the multi-image scenario for future work.

[2] We drop the distinction between support and query set for better readability in this section.

and gives us hidden visual features Z_v. These are then passed to an attention-mapper M_θ to distill task-relevant image features from Z_v.

Attention Mapper It takes a concatenated sequence of embeddings $C = (P, Z_v)$, where P represents m learnable prompt tokens (see Figure 1). The mapper then computes the corresponding query, key and value vectors which are passed to a softmax function to compute activation scores for every feature in the value vector. Finally, we extract the first m embeddings from the output to get the task-specific image embeddings H_p. We denote the trainable parameters for the attention-mapper with $\theta_p = \{\theta, P\}$.

Language Model The quality of the learned prompts highly depends on the underlying language model. To this end, we employ the state-of-the-art Qwen2.5-7B-Instruct LLM, which has demonstrated strong performance on complex tasks such as mathematical reasoning and coding and supports the generation of up to 8K tokens. The LLM (f_ϕ) receives the concatenated sequence of image and text tokens to generate the answer $X_a = f_\phi([H_p, H_q])$.

In this pipeline, we only train the attention mapper parameters θ_p which makes our approach parameter-efficient for cross-task generalization. The number of trainable parameters is approximately 24M (see Appendix A.3 for hyperparameters) and the training objective maximizes the likelihood function, $p_{\theta_p}(X_a|X_v, X_q)$. For clarity, we refer to this model as LLaVA-ATT-Qwen2.5 in the following sections.

2.3. Model Training

Similar to LLaVA v1.5 (Liu et al., 2024), we first train the attention-mapper parameters θ_p during the *pretraining* stage on LCS-558K subset (Liu et al., 2023). In the *finetuning* stage, which aims to distill task-specific image features into prompts H_p, we finetune θ_p on diverse task-specific instructions. We describe our MAML-based finetuning procedure below along with alternative methods which we compare against in our experiments.

2.3.1. Distilling Prompts with First-order Meta-learning

We refer to our approach as **Meta-Learned$^{\textbf{PD}}$** and use the implementation of (Antoniou et al., 2019) with their first-order version to finetune the attention-mapper parameters over a batch B of meta-tasks. The inner-loop step uses the support set of each task in B to convert meta-parameters θ_p into task-specific parameters θ_p'.

$$L_{\theta_p}^{\text{supp}} = \frac{-1}{|D^{\text{supp}}|} \sum_{i=1}^{|D^{\text{supp}}|} \log(p_{\theta_p}(X_a^i|X_v^i, X_q^i)) \tag{1}$$

$$\theta_p' = \theta_p - \alpha \nabla_{\theta_p} L_{\theta_p}^{\text{supp}} \tag{2}$$

The *outer* loop involves optimizing the meta-parameters on the query set using the task-specific parameters θ_p':

$$L_{\theta_p'}^{\text{query}} = \frac{-1}{|D^{\text{query}}|} \sum_{i=1}^{|D^{\text{query}}|} \log(p_{\theta_p'}(X_a^i|X_v^i, X_q^i)) \tag{3}$$

$$\theta_p := \theta_p - \beta \sum_{j=1}^{|B|} \nabla_{\theta_{p,j}'} L_{\theta_{p,j}'}^{\text{query}} \tag{4}$$

Equation (4) is the first-order approximation of the meta-update in MAML (Finn et al., 2017) that treats the gradient of $\theta_{p,j}'$ w.r.t. θ_p for a meta-task as a constant.

2.3.2. Other Prompt Distillation Methods

- **Multi-Task Prompt Distillation (Multi-Task$^{\textbf{PD}}$)** This involves distilling prompts by getting rid of the bi-level optimization of Meta-Learned$^{\text{PD}}$ and optimizing the below loss per task.

$$L_{\theta_p} = \frac{-1}{N} \sum_{i=1}^{N} \log(p_{\theta_p}(X_a^i|X_v^i, X_q^i)) \tag{5}$$

  such that $N = |D^{\text{supp}}| + |D^{\text{query}}|$.

- **In-Context Prompt Distillation (In-Context$^{\textbf{PD}}$)** Inspired from previous works (Chen et al., 2022) that reduce meta-learning of task information to a sequence prediction problem, we develop this approach and concatenate the support set with the query example to optimize the below loss for prompt distillation

$$L_{\theta_p} = \frac{-1}{|D^{\text{query}}|} \sum_{i=1}^{|D^{\text{query}}|} \log(p_{\theta_p}(X_a^i|X_v^i, X_q^i, D^{\text{supp}})) \tag{6}$$

- **Methods without Meta-tasks** We further compare with methods that do not involve any meta-tasks during training, (a) **NoMetaTask$^{\textbf{PD}}$** that mimics the original finetuning procedure of LLaVA v1.5 (Liu et al., 2024), and (b) **Model-Avg$^{\textbf{PD}}$**, where we separately finetune the attention-mapper parameters θ_p on each dataset $D^i \sim p(\mathcal{D})$, and take an average of all dataset-specific parameters θ_p^i weighted by their corresponding dataset size ratios, $\theta_p^{\text{avg}} = \sum_{i=1}^{|\mathcal{D}|} \theta_p^i \cdot \frac{|D^i|}{|\mathcal{D}|}$

2.4. Test-time Adaptation

At test-time, we adapt the attention-mapper parameters θ_p to a new (test) task by finetuning for K gradient steps. We experiment with a range of K values and explain how we select the best one for a test task in Appendix A.5. We finetune the parameters θ_p on the support set $D_{\text{test}}^{\text{supp}}$ of test task T_{test}^{j} and evaluate model performance on the query set $D_{\text{test}}^{\text{query}}$ for the same task. Alternatively, we also compare with ICL adaptation at test-time for all methods.

3. Experimental Results

We use the recently introduced VL-ICL benchmark (Zong et al., 2025), designed to test the ICL capabilities of LMMs on various tasks like fast concept binding, multimodal reasoning, and fine-grained perception. Meta-tasks for testing are created by randomly sampling a support set from the training split of the VL-ICL datasets and a test/query set from their respective testing splits. We only report results on 4 single-image tasks: Fast Open-Ended MiniImageNet (Open-MI), CLEVR Count Induction, Operator Induction (OP_IND), and TextOCR. More details in Appendix A.6

Table 1. Comparison of different prompt distillation approaches on single-image tasks from VL-ICL Bench (Zong et al., 2025). We report the mean accuracy for different numbers of shots - $\{1, 2, 4, 5, 8\}$. FT = Finetuning, ICL = In-Context Learning, TTA= Test-Time Adaptation, MT = Meta-Task. The model used for this evaluation is LLaVA-ATT-Qwen2.5 described in Section 2.2.

Methods	MT	Open-MI	OP_IND	CLEVR	TextOCR
TTA = ICL					
+NoMeta-task$^{\text{PD}}$	✗	43.8	12.1	18.0	6.8
+Model-Avg$^{\text{PD}}$	✗	26.6	9.2	7.6	2.8
+In-Context$^{\text{PD}}$	✓	51.1	20.6	24.1	23.8
+Multi-Task$^{\text{PD}}$	✓	48.6	10.0	12.5	6.9
+Meta-Learned$^{\text{PD}}$	✓	53.3	9.6	12.3	7.3
TTA = FT					
+NoMeta-task$^{\text{PD}}$	✗	68.0	38.8	25.8	22.5
+Model-Avg$^{\text{PD}}$	✗	63.1	40.0	29.1	21.5
+In-Context$^{\text{PD}}$	✓	64.5	30.9	30.9	18.9
+Multi-Task$^{\text{PD}}$	✓	74.6	45.1	29.9	22.9
+Meta-Learned$^{\text{PD}}$	✓	**77.9**	**47.7**	**31.4**	**26.4**

Prompt distillation improves task induction in LMMs at test time Results from Table 1 show that FT adaptation with few-shots largely outperforms ICL at test-time with an average increase of 21.2% over all datasets. These results highly support our hypothesis that distilling task-specific information from image embeddings to create targeted prompts improves the few-shot capabilities in LMMs.

Learning using meta-tasks is beneficial for few-shot adaptation. We further see in Table 1 that for both the test-time adaptation procedures (ICL and FT), methods which are meta-task aware are indeed superior. For ICL-based adaptation, In-Context$^{\text{PD}}$ performs best, while for FT-based adaptation, our proposed approach, Meta-Learned$^{\text{PD}}$, achieves the best overall performance across all four datasets. This suggests that learning meta-tasks during training by creating batches with equal examples per task avoids overfitting to a single task.

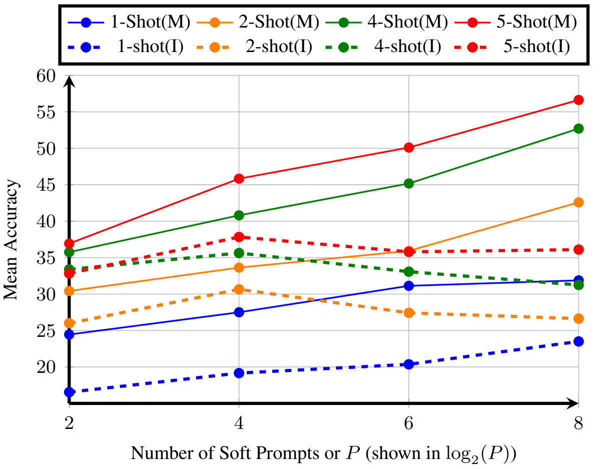


Figure 2. Performance comparison between M=Meta-Learned$^{\text{PD}}$ + FT and I=In-Context$^{\text{PD}}$+ICL. Mean Accuracy is computed across all VL-ICL datasets. We consider different prompt token lengths m or $|P| = \{4, 16, 64, 256\}$ which are shown in $\log_2(|P|)$ scale for different shots.

Meta-learning improves few-shot learning for FT-based adaptation. Table 1 also shows that our proposed meta-learning method, Meta-Learned$^{\text{PD}}$, achieves the best performance when finetuned at test-time across all datasets. This suggests that first-order MAML learns the best initialization of attention-mapper parameters θ_p which when finetuned over few-shots, are able to distill task-specific image features into soft prompts. Detailed results in Appendix A.1 show that Meta-Learned$^{\text{PD}}$ also exhibits strict monotonic improvements for all VL-ICL datasets.

Meta-Learned$^{\text{PD}}$ benefits from the addition of soft prompts in contrast to In-Context$^{\text{PD}}$. We compare Meta-Learned$^{\text{PD}}$ (the best FT approach) against In-Context$^{\text{PD}}$ (the best ICL approach) across all VL-ICL datasets, as the number of soft prompts P increases (under different shot scenarios). Figure 2 shows that Meta-Learned$^{\text{PD}}$ shows monotonically increasing performance with additional prompts. Furthermore, its marginal improvement per added prompt token is substantially greater when more shots are provided. In contrast, the performance of In-Context$^{\text{PD}}$ generally deteriorates with more prompts, except for 1-shot.

4. Conclusion

We introduce Meta-Learned$^{\text{PD}}$, a meta-learning approach that distills task-relevant image features in a fixed set of soft prompts and can induce few-shot capabilities in LMMs with finetuning-based test-time adaptation. Evaluation results on the VL-ICL benchmark suggest that Meta-Learned$^{\text{PD}}$ outperforms other ICL and prompt-tuning approaches on various VQA tasks and exhibits strictly monotonic improvements across varying number of shots and soft prompts.

5. Acknowledgements

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1).

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EbMuimAbPbs.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Antoniou, A., Edwards, H., and Storkey, A. How to train your MAML. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJGven05Y7.

Chen, Y., Zhong, R., Zha, S., Karypis, G., and He, H. Meta-learning via language model in-context tuning. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 719–730, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.53. URL https://aclanthology.org/2022.acl-long.53/.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Griffiths, T. L., Callaway, F., Chang, M. B., Grant, E., Krueger, P. M., and Lieder, F. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29:24–30, 2019. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2019.01.005. URL https://www.sciencedirect.com/science/article/pii/S2352154618302122. Artificial Intelligence.

Hendel, R., Geva, M., and Globerson, A. In-context learning creates task vectors. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Huang, B., Mitra, C., Karlinsky, L., Arbelle, A., Darrell, T., and Herzig, R. Multimodal task vectors enable many-shot multimodal in-context learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=W0okTgsPvM.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Kirsch, L. and Schmidhuber, J. Self-referential meta learning. In *First Conference on Automated Machine Learning (Late-Breaking Workshop)*, 2022. URL https://openreview.net/forum?id=WAcLlCixQP7.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., and Li, C. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=w0H2xGHlkw.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024.

Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KUNzEQMWU7.

Najdenkoska, I., Zhen, X., and Worring, M. Meta learning to bridge vision and language models for multimodal few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3oWo92cQyxL.

Qin, C., Joty, S., Li, Q., and Zhao, R. Learning to initialize: Can meta learning improve cross-task generalization in prompt tuning? In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11802–11832, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.659. URL https://aclanthology.org/2023.acl-long.659/.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJY0-Kcll.

ShareGPT. Sharegpt. https://sharegpt.com/, 2023.

Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., and Hassner, T. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8802–8812, 2021.

Tsimpoukelli, M., Menick, J. L., Cabi, S., Eslami, S. M. A., Vinyals, O., and Hill, F. Multimodal few-shot learning with frozen language models. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 200–212. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/01b7575c38dac42f3cfb7d500438b875-Paper.pdf.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3637–3645, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Zong, Y., Bohdal, O., and Hospedales, T. VL-ICL bench: The devil in the details of multimodal in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cpGPPLLYYx.

A. Appendix

A.1. Detailed Results

Table 2. Comparison of different prompt distillation approaches on single-image tasks from VL-ICL Bench (Zong et al., 2025). We report accuracy for different numbers of shots (–S). ”Avg” is only calculated for ≥ 1 shot(s). FT = Finetuning, ICL = In-Context Learning, TTA= Test-Time Adaptation. We use a maximum of $K = 30$ inner-loop gradient steps for FT adaptation (test-time). We do not compare on 0-shot results. The model used for this evaluation is LLaVA-ATT-Qwen2.5 which is described in Section 2.2.

Methods	Meta Task	Open-MI (2-way)						Operator Induction					
		0-S	1–S	2–S	4–S	5-S	Avg	0-S	1–S	2–S	4–S	8–S	Avg
TTA = ICL													
+NoMeta-task$^{\text{PD}}$	✗	0.0	35.0	47.0	48.0	45.0	43.8	11.7	13.3	13.3	10.0	11.7	12.1
+Model-Avg$^{\text{PD}}$	✗	0.0	20.0	22.0	30.0	34.5	26.6	8.3	11.7	6.7	8.3	10.0	9.2
+In-Context$^{\text{PD}}$	✓	0.0	30.0	56.0	55.0	63.5	51.1	10.0	20.0	18.5	18.0	26.0	20.6
+Multi-Task$^{\text{PD}}$	✓	0.0	43.0	50.0	51.0	50.5	48.6	8.3	13.3	11.7	3.3	11.7	10.0
+Meta-Learned$^{\text{PD}}$	✓	0.0	42.5	53.0	57.0	60.5	53.3	15.0	13.3	13.3	1.7	10.0	9.6
TTA = FT													
+NoMeta-task$^{\text{PD}}$	✗	0.0	21.5	67.5	89.0	94.0	68.0	11.7	26.7	23.3	46.7	58.3	38.8
+Model-Avg$^{\text{PD}}$	✗	0.0	28.5	53.5	83.0	87.5	63.1	8.3	31.5	28.0	45.0	55.5	40.0
+In-Context$^{\text{PD}}$	✓	0.0	35.5	54.5	79.5	88.5	64.5	10.0	21.7	18.3	41.7	41.7	30.9
+Multi-Task$^{\text{PD}}$	✓	0.0	37.0	73.5	93.5	94.5	74.6	8.3	31.0	28.3	**61.0**	60.0	45.1
+Meta-Learned$^{\text{PD}}$	✓	0.0	**43.5**	**78.0**	**94.5**	**95.5**	**77.9**	15.0	**32.0**	**38.3**	58.3	**62.0**	**47.7**
Methods	Meta Task	CLEVR Count Induction						TextOCR					
		0–S	1–S	2–S	4–S	8-S	Avg	0-S	1–S	2–S	4–S	8–S	Avg
TTA = ICL													
+NoMeta-task$^{\text{PD}}$	✗	0.0	8.0	10.5	23.0	30.5	18.0	20.0	4.5	9.5	8.5	4.5	6.8
+Model-Avg$^{\text{PD}}$	✗	1.5	17.0	8.5	4.0	1.0	7.6	12.0	3.0	2.5	3.0	1.0	2.8
+In-Context$^{\text{PD}}$	✓	0.0	13.5	23.0	28.5	31.5	24.1	16.0	22.5	21.0	23.5	28.0	23.8
+Multi-Task$^{\text{PD}}$	✓	1.0	5.0	9.0	16.5	19.5	12.5	18.0	4.0	4.5	8.5	10.5	6.9
+Meta-Learned$^{\text{PD}}$	✓	2.0	11.0	7.0	15.5	15.5	12.3	21.5	5.5	7.0	8.0	8.5	7.3
TTA = FT													
+NoMeta-task$^{\text{PD}}$	✗	0.0	18.5	21.5	26.0	37.0	25.8	20.0	20.5	23.0	24.0	22.5	22.5
+Model-Avg$^{\text{PD}}$	✗	1.5	**26.5**	25.0	29.5	35.5	29.1	12.0	17.5	20.0	23.0	25.5	21.5
+In-Context$^{\text{PD}}$	✓	0.5	24.5	**30**	**34.5**	34.5	30.9	16.0	16.0	18.0	19.5	22.0	18.9
+Multi-Task$^{\text{PD}}$	✓	0.0	25.0	25.5	31.0	38.0	29.9	18.0	21.0	20.5	24.5	25.5	22.9
+Meta-Learned$^{\text{PD}}$	✓	0.0	**26.5**	27.5	31.0	**40.5**	**31.4**	21.5	**23.5**	**26.5**	**27.0**	**28.5**	**26.4**

A.2. Finetuning Data Mixture

For model finetuning, we create our multi-task data mixture using the visual instruction tuning data of LLaVA v1.5 (Liu et al., 2023) which contains a mixture of 12 different datasets[3] ranging from long conversations to academic multiple-choice questions. Since we are only training image-based prompts, we remove the language-only ShareGPT-40K dataset (ShareGPT, 2023). Additionally, we include 3 different math reasoning/QA datasets from the LLaVA OneVision data mixture (Li et al., 2025) which are known to improve LMM performance on difficult reasoning and logical QA tasks (Lu et al., 2024). We further get rid of the extra answer formatting instructions to test the true few-shot transfer learning ability of our approach without the need of external task induction. Table 3 shows the list of all the datasets along with their size and question types.

[3] We use these datasets only for academic research purposes as mentioned by the original authors and follow the Open AI Usage Policy for GPT-4 generated datasets. Additionally, we conform to the license (CC-BY-4.0) for Cauldron datasets.

Table 3. Finetuning Data Mixture Statistics

Dataset	No. of examples	Question Types
LLaVA-Instruct	157,712	Conversations (57,669) Detailed Image Description (23,240) Complex Reasoning (76,803)
GQA	72,140	Visual Reasoning
OCR-VQA	80,000	Image Question Answering with Reading Comprehension
TextVQA	21,953	Image Question Answering with Reading Comprehension
Visual Genome	86,417	Image Question Answering and Bounding Box Prediction
MAVIS-Math-Metagen	87,348	Visual Math Question Answering
TabMWP-Cauldron	22,717	Tabular Math Reasoning
RefCOCO	48,447	Image Question Answering and Bounding Box Prediction
OKVQA	8,998	Knowledge Grounded Image Question Answering
VQAv2	82,783	Image Question Answering
A-OKVQA	66,160	Multiple-Choice Question Answering
Geo-170k (QA)	67,823	Math Question Answering and Reasoning
Total	802,498	

A.3. Model Configurations

Models We use the publicly available implementation of LLaVA v1.5[4] and first-order MAML[5] to implement our baselines. Additionally, we use the pretrained model weights from Huggingface for Qwen2.5-7B-Instruct LLM[6] and the CLIP ViT-L/14-336px visual encoder[7]. The output embedding dimension size of CLIP is 1,024 and the input word embedding size of the Qwen LLM is 3,584. We set the training context length as 4096 for all baselines except for in-context baseline where it is 8,192 as it requires training with longer sequences. The attention-mapper is a single multi-head attention block with 8 heads. The token length of the soft prompt P as described in Section 2.2 for the attention mapper is set to $m = 256$. The total number of trainable parameters for our model is approximately 24M making our approach significantly parameter-efficient for finetuning.

A.4. Training Details

Pretraining stage During the pretraining stage, we only train the attention-mapper and soft prompts for 4 epochs and use a learning rate of 2e-3 with a batch size of 64 per GPU. We perform a train-validation split on the LCS-558K dataset (Liu et al., 2023) by keeping 98% of the examples for training and 2% for validation and take the checkpoint with the lowest validation loss. We use this checkpoint as our base for further task-specific finetuning.

[4] LLaVA v1.5: `https://github.com/haotian-liu/LLaVA/tree/main/llava`

[5] How to train your MAML: `https://github.com/AntreasAntoniou/HowToTrainYourMAMLPytorch`

[6] Qwen2.5-7B-Instruct: `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`

[7] CLIP ViT-L/14-336px: `https://huggingface.co/openai/clip-vit-large-patch14-336`

Finetuning stage For finetuning, in order to keep a balanced ratio of train-validation splits across multiple datasets in Section A.2 used in this stage, we divide each dataset into 98% for training and 2% for validation and then combine them separately to create the final train and validation splits. We experimented among three different learning rates [1e-3, 5e-4, 2e-5]. For Meta-Learned$^{\text{PD}}$, we further experimented with three different inner-loop learning rates [1e-1, 5e-2, 5e-1]. Below, we mention the best learning rates along with other hyperparameters, chosen using our validation set for the different approaches proposed in Section 2.3. All approaches were finetuned for 1 epoch to ensure a complete pass over the entire finetuning data mixture.

1. **Meta-Learned$^{\text{PD}}$:** We use 5 inner-loop steps and initialize the inner-loop learning rate α=1e-1. The outer-loop learning rate is set as 1e-3 with a per GPU batch size of 1 meta-task with a gradient accumulation of 2 steps. Each meta-task here contains 10 support and 10 query examples. Training time $\sim$ 10 hours.

2. **Multi-Task$^{\text{PD}}$:** Similar to Meta-Learned$^{\text{PD}}$, we use a learning rate of 1e-3 with a per GPU batch size of 1 meta-task with a gradient accumulation of 4 steps. Each meta-task here contains 5 support and 5 query examples. Training time $\sim$ 4.5 hours

3. **In-Context$^{\text{PD}}$:** We use a learning rate of 1e-3 with a gradient accumulation of 4 steps and 5 meta tasks per GPU. Each meta task contains 10 support examples and 1 query example. The support examples were concatenated with the strategy that ensured all image tokens of a meta-task are present in the sequence and we truncate the text tokens if the sequence exceeded the context length of 8192. Further, the few-shot question and answers were concatenated by inserting ”Question:” and ”Answer:” strings in between them, inspired from (Alayrac et al., 2022). Training time $\sim$ 4.5 hours

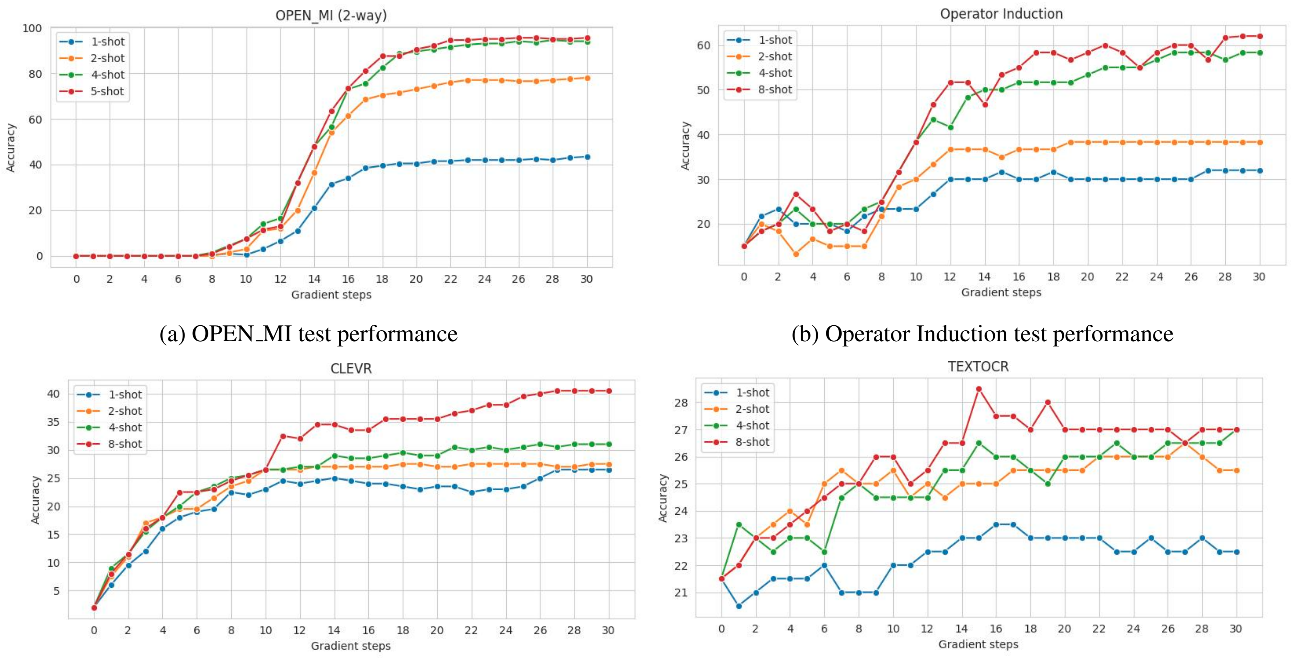
4. **Model-Avg$^{\text{PD}}$:** We first finetune individual models on each dataset in the finetuning data mixture (Section A.2) with a learning rate of 5e-4. For all the datasets, we choose a per GPU batch size of 8 with gradient accumulation of 2 steps. Average time per dataset $\sim$ 3 hours

5. **NoMeta-task$^{\text{PD}}$:** Here, we finetune on the complete data mixture in one training run by sampling batches randomly and again use a per GPU batch size of 8 with a gradient accumulation of 2 steps. We also use a learning rate of 5e-4. Training time $\sim$ 4 hours.

Computational Requirements For the entire model training, we use 4 H200 GPUs with a VRAM of 143GB per GPU. For both the stages, the hyperparameters were tuned using their corresponding validation sets and we choose the checkpoints at the end of first epoch to report our results.

A.5. Test-Time Adaptation Details

We choose a similar test-time adaptation procedure as (Qin et al., 2023) to find the best hyperparameter settings for every prompt distillation method for fair comparison. We first sample 10% of the examples from the training split of each test task and combine them to create a validation set. After meta-task creation of VL-ICL datasets (Zong et al., 2025) using the remaining training and test splits, we then performed a maximum of $K = 30$ inner-loop gradient steps over each support set of a meta-task and chose the Kth-step model that gave the lowest validation loss. We use this model to calculate the result over the query set. To further show how the performance varies at different gradient steps, we plot the average test accuracy curves for different VL-ICL datasets for Meta-Learned$^{\text{PD}}$ for different shots in Figure 3. We see that the accuracies converge or start decreasing under 30 gradient steps which validates our adaptation procedure designed to achieve best performance. We also provide examples of how the predictions change during test-time adaptation in Figure 4, Figure 5, Figure 6, and Figure 7. Further to ensure reproducibility, we provide our best learning rate values in Table 4 used for different methods based on the validation set after doing a hyperparameter search within the range $[0.1, 1.0]$ with a batch size of 1 meta-task.



(a) OPEN_MI test performance

(b) Operator Induction test performance

(c) CLEVR test performance

(d) TEXTOCR test performance

Figure 3. Average test performances of Meta-Learned$^{\text{PD}}$ with finetuning on different datasets

Table 4. Learning rates for finetuning-based (FT) test-time adaptation for results shown in Table 1 and Table 2

Training Methods	Learning Rate (LR)
Meta-Learned$^{\text{PD}}$	1.0
Multi-Task$^{\text{PD}}$	0.8
In-Context$^{\text{PD}}$	0.8
ModelAvg$^{\text{PD}}$	0.6
NoMeta-task$^{\text{PD}}$	1.0

A.6. Evaluation Datasets from VL-ICL Bench

The VL-ICL Bench (Zong et al., 2025) includes a diverse variety of tasks to test different capabilities of models like Fast-Concept binding, Mathematical Induction, and Fine-grained perception. Given the nature of our model architecture and training (Section 2.2, Section 2.3), we only focus on the single-image Image-to-text (I2T) tasks.

1. **Fast Open-Ended MiniImageNet (OPEN_MI)** - This is a variant of the MiniImageNet few-shot object recognition task (Vinyals et al., 2016), which was repurposed for few-shot prompting (Tsimpoukelli et al., 2021). It is essentially an open-ended image classfication problem, but contains nonsense categorical names like *dax* or *blicket* making the test performance not influenced by the prior knowledge of an LMM but only dependent on the support examples. This design ensures to test the few-shot abilities of LMMs and how quickly they can learn about new concepts. For the results shown in Table 2, we use the 2-way version of this task involving classification between two nonsense categories. An example of a 2-way 1-shot task is shown in Figure 4.

2. **Operator Induction** - Initially proposed by (Zong et al., 2025), this dataset tests various capabilties of LMMs like Task Induction, Perception and Mathematical Reasoning. The support examples involve two operands with a missing mathematical operation and an answer. When testing, the task is to identify the hidden operation from the support example and use it to calculate the result over the operands in the query. An example of a 2-shot task is shown in Figure 7.

3. **CLEVR Count Induction** - This dataset contains images from the widely used CLEVR dataset (Johnson et al., 2017) where each image contains a set of objects that have certain characteristics based on attributes like shape, size, color and material. The task is to learn to count the objects of the given attribute in the support example and transfer that knowledge to count the objects of any attribute in the query example. An example of a 2-shot task is shown in Figure 5.

4. **TextOCR** - This dataset has been repurposed by (Zong et al., 2025) from the TextOCR dataset (Singh et al., 2021) to create a task where the LMM should learn to output the text within a red bounding box from the support examples. Even though this task could be solved in a zero-shot setting as we see in the 0-shot case with a detailed prompt, we still only focus on inducing task knowledge from the few-shot examples. An example of a 2-shot task is shown in Figure 6.

A.7. Qualitative Results

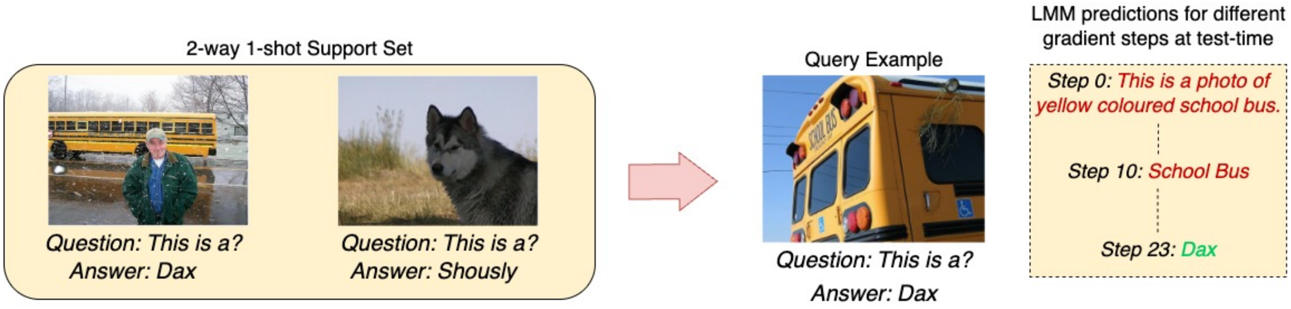


Figure 4. OPEN_MI predictions at test-time

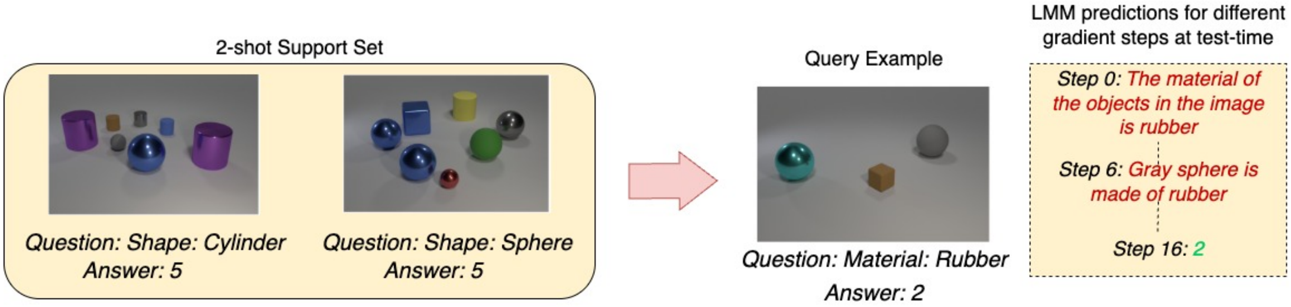

2-shot Support Set
Question: Shape: Cylinder
Answer: 5
Question: Shape: Sphere
Answer: 5
Query Example
Question: Material: Rubber
Answer: 2
LMM predictions for different gradient steps at test-time
Step 0: The material of the objects in the image is rubber
Step 6: Gray sphere is made of rubber
Step 16: 2

Figure 5. CLEVR predictions at test-time

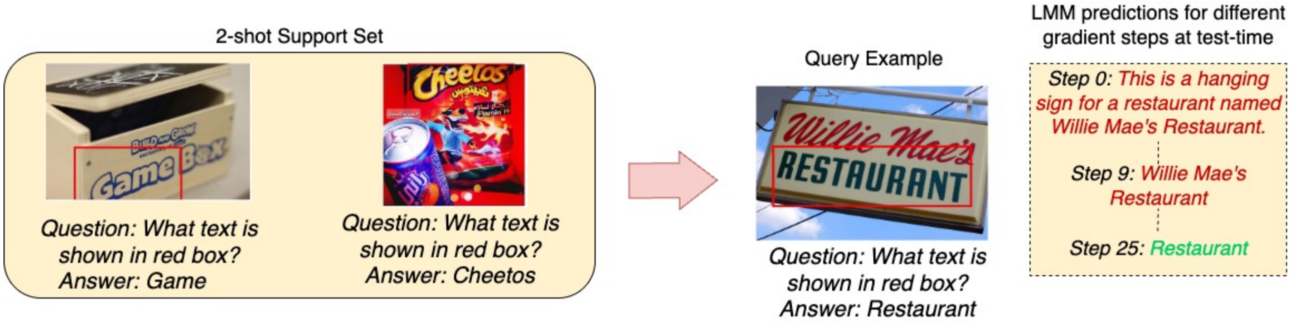

2-shot Support Set
Question: What text is shown in red box?
Answer: Game
Question: What text is shown in red box?
Answer: Cheetos
Query Example
Question: What text is shown in red box?
Answer: Restaurant
LMM predictions for different gradient steps at test-time
Step 0: This is a hanging sign for a restaurant named Willie Mae's Restaurant.
Step 9: Willie Mae's Restaurant
Step 25: Restaurant

Figure 6. TEXTOCR predictions at test-time

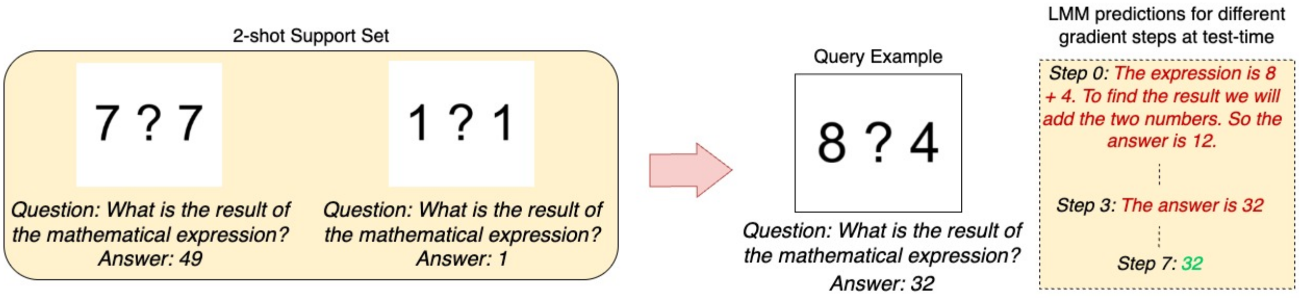

2-shot Support Set
7 ? 7
Question: What is the result of the mathematical expression?
Answer: 49
1 ? 1
Question: What is the result of the mathematical expression?
Answer: 1
Query Example
8 ? 4
Question: What is the result of the mathematical expression?
Answer: 32
LMM predictions for different gradient steps at test-time
Step 0: The expression is 8 + 4. To find the result we will add the two numbers. So the answer is 12.
Step 3: The answer is 32
Step 7: 32

Figure 7. Operator Induction predictions at test-time